



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/040,578

12/28/2001

David J. Long

50277-1766

3641

42425

7590

03/16/2009

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE, CA 95110-1083

EXAMINER

PHAM, CHRYSTINE

ART UNIT

PAPER NUMBER

2192

MAIL DATE

DELIVERY MODE

03/16/2009

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte DAVID J. LONG and DAVID B. PITFIELD
Appeal 2008-2928
Application 10/040,578¹
Technology Center 2100

Decided:² March 16, 2009

Before ALLEN R. MACDONALD, JEAN R. HOMERE, and
STEPHEN C. SIU, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

¹ Filed on December 28, 2001. The real party in interest is Oracle International Corp.

² The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, begins to run from the decided date shown on this page of the decision. The time period does not run from the Mail Date (paper delivery) or Notification Date (electronic delivery).

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1 through 32. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

Appellants' Invention

As shown in Figures 2a and 7a, Appellants invented a computer-implemented method for associating an instance (700) of a class (220) with an attribute that is not in the class (220) or in a superclass (210) thereof, and wherein the association does not cause the attribute to become part of the class. (Spec. 15, ll. 11-16, l. 2, *id.* 22, ll. 12-21.)

Illustrative Claim

Independent claim 1 further illustrates the invention. It reads as follows:

1. A method of specifying a structure, within a computer system, of an instance of a class, the method including the step of:

associating with said instance of said class an attribute that is not in said class or any superclass of said class, thereby establishing for said instance said structure that includes storage for data associated with said attribute; and

wherein associating said attribute with said instance does not cause said attribute to become an attribute of said class.

Appeal 2008-2928
Application 10/040,578

Prior Art Relied Upon

The Examiner relies on the following prior art as evidence of unpatentability:

Ng	US 6,385,618 B1	May 7, 2002 (filed Jun. 29, 1998)
----	-----------------	--------------------------------------

The JavaTM Virtual Machine Specification, *The class File Format*, Sun Microsystems, Inc., (1996, 1997) available at <http://java.sun.com/docs/books/vmspec/html/ClassFile.doc.html>

Rejections on Appeal

The Examiner rejects the claims on appeal as follows:

1. Claims 1 through 15 and 17 through 31 stand rejected under 35 U.S.C. § 102b(e) as being anticipated by Ng.
2. Claims 16 and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of Ng and the Java Virtual Machine Specification reference.

Appellants' Contentions

Appellants argue that Ng's disclosure does not teach associating with an instance of a class an attribute that is not in the class or a superclass thereof such that the attribute does not become part of the class. (App. Br. 6-12, Reply Br. 2-4.) Particularly, Appellants argue that while the "collection Orders for Customer" is an attribute of the Orders table, it is also a part of the Customers class by dint of its association with an instance

(customer_id) of the customer class. (*Id.*) Consequently, Appellants submit that Ng does not anticipate independent claim 1.

Examiner's Findings/Conclusions

The Examiner finds that Ng's "collection of Orders_for_Customer" is a collection of objects representing one or more instances in the Orders table for a particular customer in the Customer table. (Ans. 4.) The Examiner also finds that while the collection of "Orders_for_Customer" is itself an attribute of the Customer class, the orders contained in the collection are instances of an attribute in the Order class, and not in the Customer class or superclass. That is, the attribute (orders) of the Order class is associated with an instance of the Customer class (a particular, i.e. customer_id.), and that such an association does not make the Orders attribute a part of the Customer class. Consequently, the Examiner finds that Ng anticipates independent claim 1. (*Id.*)

II. ISSUE

Have Appellants shown that the Examiner erred in finding that Ng's disclosure anticipates the claimed invention? Particularly, the issue turns on whether Ng's disclosure of a "collection of Orders_for_Customer" teaches associating with an instance of a class an attribute that is not in the class or a superclass thereof such that the associated attribute does not become part of the class, as recited in independent claim 1.

III. FINDINGS OF FACT

The following findings of fact (FF) are supported by a preponderance of the evidence.

Ng

1a. As depicted in Figure 1, Ng discloses a computer (101) having an object-relational mapping tool (ORMT, 114) that queries a database (118) to identify its schema. The ORMT (114) then uses the identified schema to construct a database data structure (115). The ORMT (114) uses the database data structure (115) to generate an object model (116). The ORMT (114) allows a programmer to customize the object model (116) (e.g. by adding/removing fields thereto), and subsequently uses the customized object model to create a source code (120). (Col. 5, ll. 23-36.)

1b. As shown in Figure 4B, Ng discloses a source code file (116) generated to represent two classes (420, 424) of objects (400 and 401) reflecting Customer table (202) and Order table (204) respectively. (Col. 6, ll. 32-38.)

1c. The Customer class (420) includes *inter alia* customer_id attribute, as well as a collection Orders_for_Customer attribute, representing the orders associated with a particular customer. As shown in Figure 4B, the collection Orders_for_Customer attribute includes commands to identify a customer as well as to obtain orders for the identified customer (get_Cust_id (); getOrdersForCustomer ()). (Col. 6, ll. 38-42.)

1d. The Order class (424) includes *inter alia* customer_id attribute, Order data, as well as Customer_for_ Order attribute, implementing a foreign key referencing a particular customer that placed an order. (Col. 6, ll. 42-50.)

1e. Ng defines collection as a type indicating a grouping of instances of other classes. (Col. 6, ll. 59-64.)

IV. PRINCIPLES OF LAW

Anticipation

In rejecting claims under 35 U.S.C. § 102, “[a] single prior art reference that discloses, either expressly or inherently, each limitation of a claim invalidates that claim by anticipation.” *Perricone v. Medicis Pharmaceutical Corp.*, 432 F.3d 1368, 1375 (Fed. Cir. 2005), citing *Minn. Mining & Mfg. Co. v. Johnson & Johnson Orthopaedics, Inc.*, 976 F.2d 1559, 1565 (Fed. Cir. 1992). “Anticipation of a patent claim requires a finding that the claim at issue ‘reads on’ a prior art reference.” *Atlas Powder Co. v. IRECO, Inc.*, 190 F.3d 1342, 1346 (Fed Cir. 1999) (“In other words, if granting patent protection on the disputed claim would allow the patentee to exclude the public from practicing the prior art, then that claim is anticipated, regardless of whether it also covers subject matter not in the prior art.”) (internal citations omitted).

V. ANALYSIS

Independent claim 1 recites in relevant part associating with an instance of a class an attribute that is not in the class or a superclass thereof such that the associated attribute does not become part of the class.

As set forth in the Findings of Facts section, Ng discloses a source code file including a Customer class having a customer id, and a collection Orders_for_Customer attribute designating a collection of orders for a particular customer. (FF. 1c.) The source code file also includes a separate Order class containing order data pertaining to each identified customer. (FF. 1d.) Ng defines a collection as a grouping of instances of another class. (FF. 1e.) We agree with the Examiner that the collection Orders_for_Customer attribute reasonably teaches the afore-cited limitation. Specifically, we find that Ng's collection of orders teaches a plurality of instances of an attribute in the Order class, wherein the cited attribute (or the instances thereof) is associated with a particularly identified customer entry, which is an instance in the Customer class. Further, we find that the order attribute (or the instances thereof) is not in the Customer class or a superclass of the customer instance associated therewith. Plainly put, we find that Ng teaches associating with a customer_id in the Customer class an order attribute (or instances thereof) in the Order class, wherein the attribute order (or instances thereof) is not in the Customer class or a superclass thereof. While the customer_id attribute is shared between the Customer class and the Order class, the Order attribute (or the instances thereof)

remains a part of the Order class, and it is never a part of the Customer class. The order attribute (or the instances thereof) must therefore be fetched from the Order table when a command (getOrdersForCustomer()) is received to gather such information for a particular customer. (FF. 1c.) Consequently, we find that Ng also reasonably teaches that the association of the order attribute with the customer instance does not cause the attribute to be part of the Customer class. It follows that Appellants have not shown that the Examiner erred in finding that Ng anticipates independent claim 1.

As to claim 2, Appellants argue that Ng does not teach a property bundle associated with an attribute that is not in the same class or superclass as an instance associated therewith. (App. Br. 12-13.) We do not agree. As discussed above, Ng's order attribute includes a plurality of order instances in the Order class, wherein each identified customer in the Customer class is linked to the Order class attribute or instances thereof. Therefore, the association of the order attribute (or instances thereof) in the Order class with the customer instance in the Customer class teaches the claimed limitation.

As to claim 3, Appellants argue that Ng does not teach storing within a database objects that define the instance, the property and the attributes. (App. Br. 14-15.) We do not agree. As set forth in the Findings of Facts, the source file is a representation of data stored in the data tables of the relational database depicted in Figure 2. (FF. 1b.) Therefore, the order

attribute (or the instances thereof) is stored in the Order table and the customer associated therewith is stored in customer table.

As to claim 4, Appellants argue that Ng does not maintain an object relational mapping system indicating a correlation between the instance and data stored in the relational database. (App. Br. 15-16.) We do not agree. As discussed above, Ng discloses a plurality of tables in a relational database for storing the customer data and order data corresponding thereto. Thus, the disclosed relational database inherently teaches a correlation between the different attributes and instances of said tables.

As to claims 5 and 6, Appellants argue that Ng does not teach a pointer for linking the instance of a class with an attribute of another class. (App. Br. 16-18.) We do not agree. As discussed above, Ng discloses a plurality of relational tables that stores a plurality of classes of objects including associated attributes, instances and correlations therebetween. Thus, Ng's relational database inherently discloses pointers linking the order attribute (or the instances thereof) and the customer instance in such a way to enable the retrieval of orders corresponding to an identified customer.

Appellants did not provide separate arguments with respect to the rejection of claims 7 through 32. Appellants are reminded that merely repeating the language of the claim or separately reiterating an earlier argument does not constitute a separate argument. It follows for the same reasons detailed in our discussion of claims 1 through 6 above that

Appeal 2008-2928
Application 10/040,578

Appellants have not shown that the Examiner erred in rejecting claims 7 through 32. 37 C.F.R. § 41.37(c)(1)(vii).

VI. CONCLUSIONS OF LAW

1. Appellants have not shown that the Examiner erred in finding that Ng anticipates claims 1 through 15 and 17 through 31.

2. Appellants have not shown that the Examiner erred in concluding that the combination of Ng and Java Virtual Machine renders claims 16 and 32 unpatentable.

VII. DECISION

We affirm the Examiner's decision to reject claims 1 through 32.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rwk

HICKMAN PALERMO TRUONG & BECKER/ORACLE
2055 GATEWAY PLACE
SUITE 550
SAN JOSE CA 95110-1083